TECHNICAL REPORT

Advanced Technologies for Acoustic Monitoring of Bird Populations

SERDP Project RC-1461

APRIL 2009

Christopher W. Clark, PhD
Kurt M. Fristrup, PhD
Cornell University

SERDP
DOD ▪ EPA ▪ DOE

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **APR 2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE **Advanced Technologies for Acoustic Monitoring of Bird Populations**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Cornell University,Ithaca,NY,14853**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**26**	

Background

DoD lands are critically important to migratory bird species as breeding sites, wintering sites, and as migratory stopover sites. The Endangered Species Act (ESA) requires that US military installations monitor the status of federally listed threatened and endangered species (TES) on their grounds.

The standard approach to monitoring populations of breeding songbirds relies on point counts in which a skilled observer notes the species, and in some cases the numbers, of all birds heard or seen at a sampling point during a short (typically 3 – 10 minute) count interval. Often the majority of individual bird detections in point counts are acoustic; many of the birds noted during a typical count are never seen. In suburban landscapes, closed-canopy deciduous habitats, and tropical forested habitats, acoustic detections can comprise 70 – 94% of all detections (Alldredge et al. 2006, and references therein).

In monitoring some groups such as nocturnal birds (notably owls and nightjars) and some secretive marsh birds, virtually all detections are acoustic. In some of these species, monitoring efforts are further hampered by the birds' infrequent and unpredictable vocal activity, which may require impractically long observer times at each point in order to have reasonable confidence about the absence of a species at the site.

Because acoustic detection plays such a prominent role in avian population monitoring, the use of automated acoustical recording instruments and signal detection and classification software has the potential to lead to improved monitoring of bird populations on DoD lands and elsewhere. Specifically, such techniques may enable more extensive sampling, improved estimates of the birds counted and missed, and improved estimates of the area surveyed.

These hardware and software tools can also enable passive acoustic monitoring of nocturnally migrating birds across large geographic scales. Such migration monitoring may be especially useful in assessing the use of DoD lands as migratory stopover sites.

In April 2005 a contract (SI-1461) was awarded by SERDP to Cornell University to promote the development of "Advanced Technologies for Acoustic Monitoring of Bird Populations." The objectives of this project are to:

- Improve automated acoustic processing software to enable widespread use of digital autonomous recording units (ARUs) for:
  (a) ground-based acoustic censusing of species that vocalize infrequently,
  (b) documenting variation in calling activity to improve the accuracy of all acoustic censuses and the value of historical data sets;
- Improve and extend technology for conducting line transect surveys using free-drifting balloons, first developed under a previous SERDP contract, SI-1185;
- Develop the critical hardware and software components for a network of acoustic detectors to monitor flight calls of nocturnally migrating bird species, to document species-specific stopover use on and around DoD installations.

This document summarizes all work conducted under SI-1461 and is submitted as a final, contract closeout report.[1]

Activities and accomplishments

Acquisition of training/test data for detector development

The original proposal called for deployment of ARUs at multiple DoD bases to record audio data that would be used for training and testing of automated detectors for selected bird species of interest. Shortly after the award of SI-1461, the DoD Legacy Resource Management Program approved funding for a related proposal submitted by Dr. Kenneth Rosenberg (Cornell Lab of Ornithology) titled "Migratory Bird Monitoring Using Automated Acoustic and Internet Technologies" (Legacy Project 5-245). The Legacy-funded project included extensive deployments of ARUs at multiple bases. The decision was therefore made to use recordings collected by the Legacy project as the source of training and testing data for SI-1461, rather than expend SERDP resources to collect equivalent data independently.

Table 1 lists bases where the Legacy project collected audio data that were available to SI-1461. Additional data available for use in this project were collected by BRP projects funded by other Federal agencies including USDA Forest Service, US Fish and Wildlife Service, and US Geological Survey.

[1] The original proposal for this project, which described a four-year effort, was conceived and submitted by Dr. Kurt Fristrup, then Assistant Director of the Bioacoustics Research Program (BRP) at the Cornell Laboratory of Ornithology. At the time that SI-1461 was awarded to Cornell University (April 2005), Fristrup was named as Principal Investigator. In November 2005, mid-way through Year 1 of the project, Fristrup left Cornell to accept a position with the National Park Service Natural Sounds Program Office. At that time Dr. Christopher Clark (Director of BRP) was named as the new PI on SI-1461 for the remainder of Year 1 of the project.

Table 1. DoD sites where ARUs were deployed since September 2005 as part of the DoD Legacy migratory bird monitoring project.

DoD Legacy ARU deployment sites
Fort Drum (NY)
West Point (NY)
Picatinny Arsenal (NJ)
Lakehurst NAS (NJ)
Dover AFB (DE)
Patuxent River NAS (MD)
Camp Pendleton (CA)
Whidbey Island (WA)
Yakima Training Center (WA)
Fallon NAS (NV)
Vandenberg AFB (CA)

Table 2 lists species identified by DoD Partners In Flight (PIF) representatives as potential targets for ARU studies for which ARU recordings are known or likely to be available, either from deployments on DoD lands or elsewhere.

Table 2. Availability of ARU recordings for bird species of interest to DoD resource managers. 'Available' indicates species presence in recordings has been confirmed. 'Likely' and 'Possible' designations are based on dates and locations of recordings in relation to known distributions and habitat preferences of these species.

Availability of ARU recordings	Species
Available	Chuck-will's-widow
	Whip-poor-will
	Black-capped Vireo
	Wood Thrush
	Golden-cheeked Warbler
	Prothonotary Warbler
	Cerulean Warbler
Likely	Upland Sandpiper
	Long-billed Curlew
	Least Bell's Vireo
	Gray Vireo
	Louisiana Waterthrush
Possible	Mountain Plover
	Prairie Warbler
	Kentucky Warbler
	Grasshopper Sparrow
	Henslow's Sparrow

Detection and classification software for songs/calls of target species

The Bioacoustics Research Program has developed two interactive sound analysis software packages: *XBAT* and *Raven*. Both programs incorporate interactive sound visualization, measurement, and annotation tools. Enhancements were made to both programs under SI-1461 to improve their utility for tasks such as detecting and classifying sounds from species of interest on DoD lands.

XBAT (eXtensible Biaocoustics Tool, www.xbat.org) is an open-source program that operates within MATLAB (The MathWorks, Inc.). It is designed to be both an easily extensible platform for rapid implementation of automated tools for detecting and measuring sounds of interest, and a production environment for automated analysis of arbitrarily large acoustic data sets. Detection and measurement algorithms can be developed in MATLAB (a leading development environment for scientific and engineering software) and easily "plugged in" to the XBAT framework. The output of these automated tools can then be rapidly viewed, verified, and if necessary edited, by a human analyst working within XBAT's flexible, user-friendly visualization environment. Under SI-1461, enhancements were made to two of XBAT's sound detectors, a new database data log storage format

was implemented, and algorithms needed for efficient nearest-neighbor classification of sounds were implemented.

Raven Pro is a standalone program (it does not require MATLAB or other software) that has been licensed by over 1000 research and education professionals worldwide. Raven Pro is widely recognized for its flexible displays and analytical power, combined with an exceptionally elegant user interface. Under SI-1461, major architectural changes were implemented in Raven to make the program more easily extensible, and two automatic sound detectors were added.

These development efforts are described in more detail below.

Data template detector extension for XBAT

XBAT's *data template detector* scans a recorded sound stream and finds sounds that are similar to a detection *template* known to be from the target species. The data template detector quantifies acoustic similarity by spectrogram cross-correlation, and logs all events for which the correlation value exceeds a specified threshold.

Enhancements to data template detector

Under SI-1461, the following enhancements of XBAT's data template detector were implemented:

- **Multiple templates:** At the start of this project, the data template detector could only compare the sound stream to one template at a time. Because the sounds of most bird species are variable, this approach meant that multiple detection runs, each with a different template, were needed in order to have a high probability of finding the target sounds. Under SI-1461, the ability to run multiple templates simultaneously was added, vastly improving processing speed.
- **Rejection templates:** In some cases, templates for a particular target sound fortuitously match other sounds in a recording that are not from the intended target. If the unwanted sound recurs frequently (for example calls of a frog that happen to resemble parts of a target bird sound), the detector may generate extremely high rates of false detections. In order to mitigate this problem, a *rejection template* feature was implemented. When one or more rejection templates are specified, the detector compares all potential event detections to both the target and rejection templates. If an event is more similar to a rejection template than it is to any of the target templates, the event is rejected and not logged. Experience has shown that the use of rejection templates can reduce false detection rates by an order of magnitude.
- **Batch detection:** At the start of the project, users could only initiate a detection run on one recording at a time within the XBAT interface. Detection processing of long recordings (hundreds of hours) may take several hours, so such tasks typically run unattended. If the user needed to process multiple recordings, s/he would have to manually start each processing run after the completion of the previous run. Under SI-1461, *batch detection* capability within XBAT was implemented. Using batch detection, the user can specify an arbitrary number of recordings to process sequentially. This ability makes it possible for example to use computing time efficiently to run multiple detections sequentially overnight or over a weekend without any operator intervention.

Test applications of data template detector

Cerulean Warbler

In work partially funded by the USDA Forest Service, the data template detector (incorporating improvements made under SI-1461) was evaluated for its ability to find songs of Cerulean Warbler in ARU recordings. Cerulean Warbler (*Dendroica cerulea*, CERW) is a forest songbird of conservation concern to DoD. ARUs were deployed at eleven sites in the Allegheny National Forest in Pennsylvania. Cerulean Warblers were known to be present at four of these sites; the remaining seven sites were appropriate Cerulean Warbler habitat, but the actual presence or absence of the species at these sites was unknown. In a first-pass analysis six archived songs from the Lab of Ornithology's Macaulay Library were used as templates. The detector successfully found CERW songs on all recordings where they were known to be present (Figure 1). 66% of all detections were verified as CERW (a positive predictive value of 66%). Positive predictive value at individual recording sites varied between 39% and 77% in this first-pass analysis.

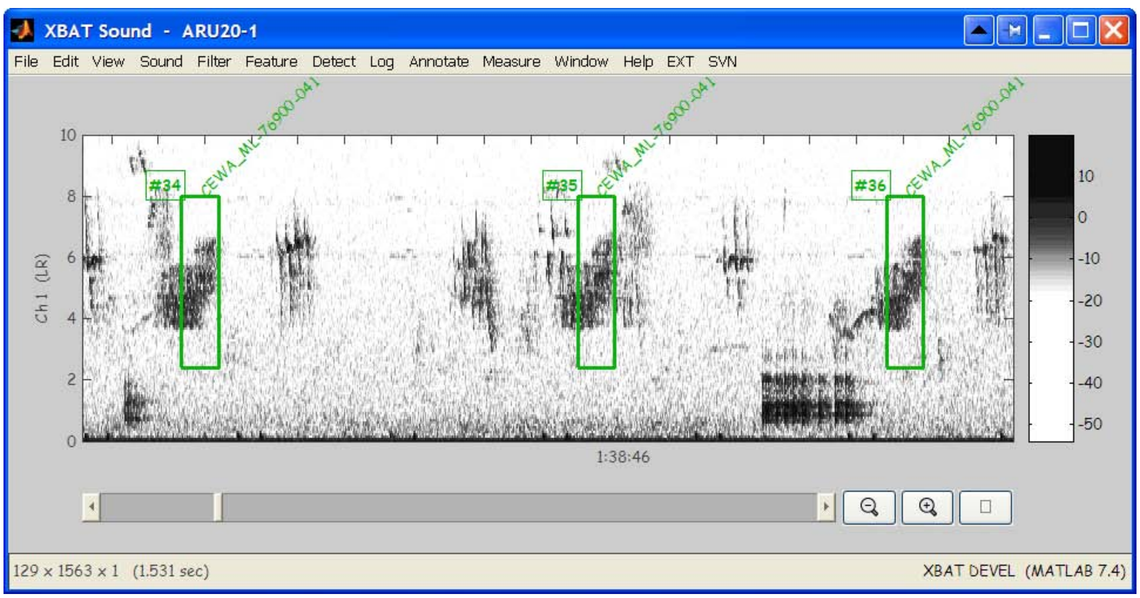

Figure 1. Sound spectrogram of 20 seconds of an ARU recording as displayed by XBAT, showing three songs of a Cerulean Warbler that were detected by the data template detector (green boxes). Other bird sounds that were ignored by the detector are visible between the marked songs.

We estimated the *sensitivity* (the percent of sounds detected out of the number of actual sounds present) of the detector by examining one five-minute sample from each morning to determine what percentage of Cerulean sounds found by a human were also found by the detector. In the first-pass analysis, the detector found 23% of songs found by a human analyst.

In a subsequent second pass, we refined the detector by eliminating templates from the first pass that did not perform well, lengthening the templates to include the entire song, adding deployment-specific templates of local song variants, and adding deployment-specific rejection templates based on what was being falsely detected on each deployment. In this second pass, positive predictive value

improved to 100% (no false detections), and estimated sensitivity improved to 54%. These improvements in performance were directly dependent on software features (multiple templates and rejection templates) implemented under SI-1461.

Whip-poor-will

In work partially funded by DoD Legacy, we investigated the utility of the data template detector for detecting sounds of Whip-poor-will (*Caprimulgus vociferus*), a nocturnal forest bird identified by DoD personnel as a species of conservation concern. We used six archived whip-poor-will song phrases from the Macaulay Library as templates. Table 3 summarizes the performance of the data template detector at various correlation threshold values. The detector successfully found WPWI phrases even when the signal-to-noise ratio (SNR) was poor because of the bird's distance from the recorder (Figure 2).

Table 3. Performance of data template detector at detecting songs of Whip-poor-will in ARU recordings from two sites at Fort Drum, NY over a 15-day period centered on the June 2007 full moon. *PPV* = estimated positive predictive value, based a sample of 1000 detections from each deployment. *Sensitivity* = estimated sensitivity based on 1000 1-minute samples from each deployment.

	Correlation threshold		
	0.30	0.20	0.15
Events detected	125,891	427,901	482,409
PPV	99.7%	98.5%	96.2%
Sensitivity	47.2%	70.3%	80.3%

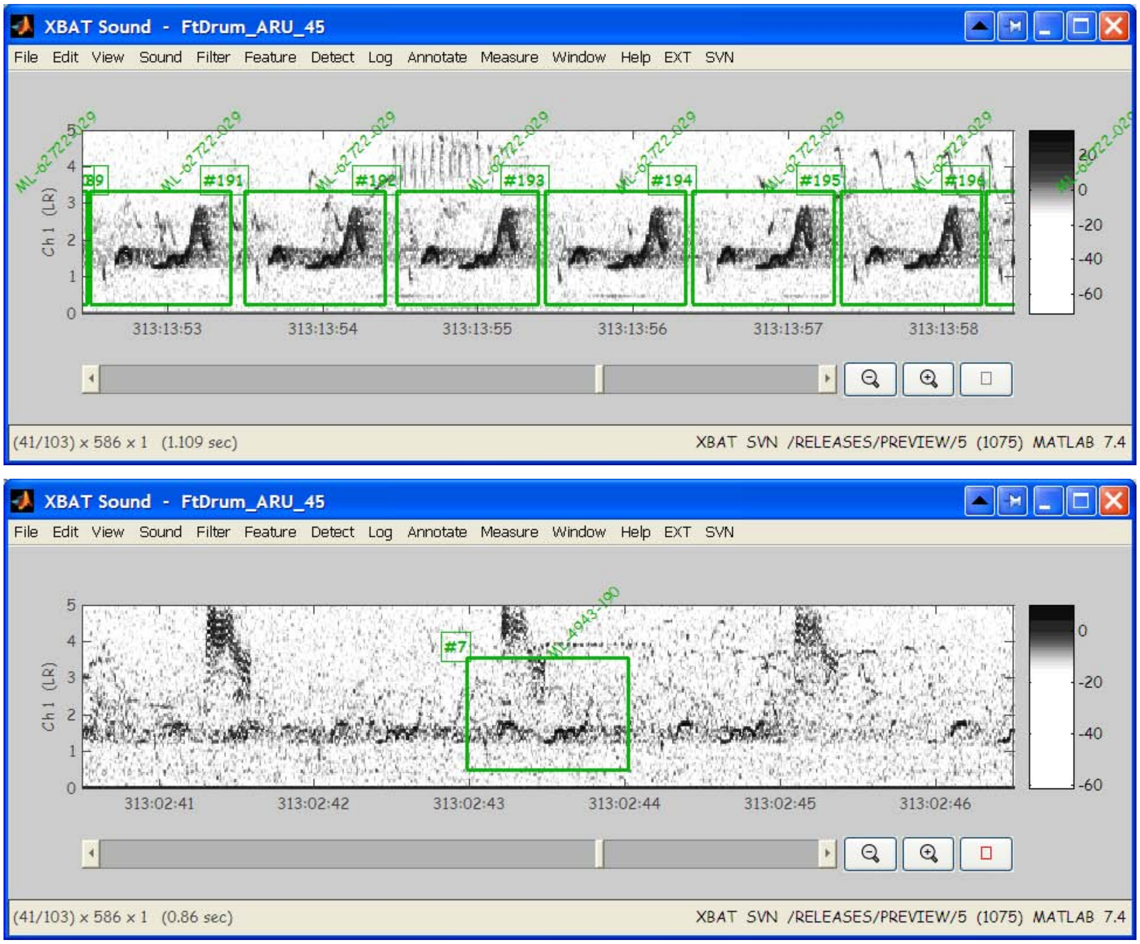

Figure 2. Detection of whip-poor-will song phrases by XBAT data template detector. **Upper:** Screenshot of XBAT sound spectrogram window illustrating consistent detection of high signal-to-noise ratio (SNR) sounds of nearby vocalizing bird. **Lower:** Detection of very faint (low SNR) sound from distant bird.

Whip-poor-wills are known to be more vocally active on nights with high levels of lunar illumination (Mills 1986; Wilson and Watts 2006). Standard protocols for monitoring whip-poor-will populations call for an observer to listen for a six-minute sample period at each survey site. Because of the low probability of detecting a whip-poor-will that is present at a site during the darker portions of the lunar cycle, standard protocols limit surveys to only two two-week periods during each spring/summer, centered around the full moons. However, the data template detector successfully identified whip-poor-will songs in ARU recordings even during the three darkest nights of the lunar cycle centered on the new moon. These results suggest that use of ARUs and automated detectors may enable monitoring of more sites on more dates than is possible using established field protocols.

Band-limited energy detector diagnostic displays for XBAT

A band-limited energy detector had been implemented as an extension to XBAT before the start of this project. Under SI-1461, the detector was enhanced with a diagnostic display that visualizes

results of the several intermediate steps in the detector algorithm, enabling the user to rapidly and efficiently configure the detector for improved performance.

The band-limited energy detector is a four-stage process:

1. The background noise in a target frequency band is estimated by computing the median Fourier spectrum for successive segments of time series data. The in-band noise power is then obtained by summing the power values in the median spectrum over the appropriate frequency bins. The block of data used to compute the median spectrum is typically many times longer than the longest sound of interest.
2. The signal power in a target frequency band is estimated for the same segments of time series data by subtracting the estimated in-band noise power (from step 1) from the overall in-band power. The in-band signal and noise estimates are used to compute a series of signal-to-noise ratio (SNR) estimates for the data.
3. Candidate detections are generated, which begin when the SNR exceeds a user-specified threshold, and which end when the SNR remains below threshold longer than a user-specified duration.
4. A candidate is marked as a valid detection if (1) the fraction of short-time SNR values above threshold is greater than a user-specified minimum "occupancy," and (2) its duration falls between a user-specified minimum and maximum.

Enhancements to band-limited energy detector

The first attempts to use the band-limited energy detector for large-scale data processing were made as part of the DoD funded Legacy nocturnal flight call project. These efforts demonstrated that it was often difficult and time-consuming to configure the detector's various parameters for acceptable performance. When the detector missed target events, or falsely detected non-targets, it was often unclear which parameter(s) (e.g., SNR threshold or minimum occupancy) to change to improve performance. In addition, changes to a single parameter could produce results in the final output of the detector that were unexpected because the results of each intermediate stage in the detection process were not observable.

Under SI-1461, a set of diagnostic displays were implemented that show the results of each intermediate stage in the detection process (Figure 3). These diagnostics take the guesswork out of detector configuration and enable the user to make targeted improvements to detector performance in a few minutes that previously could have taken hours of trial and error.

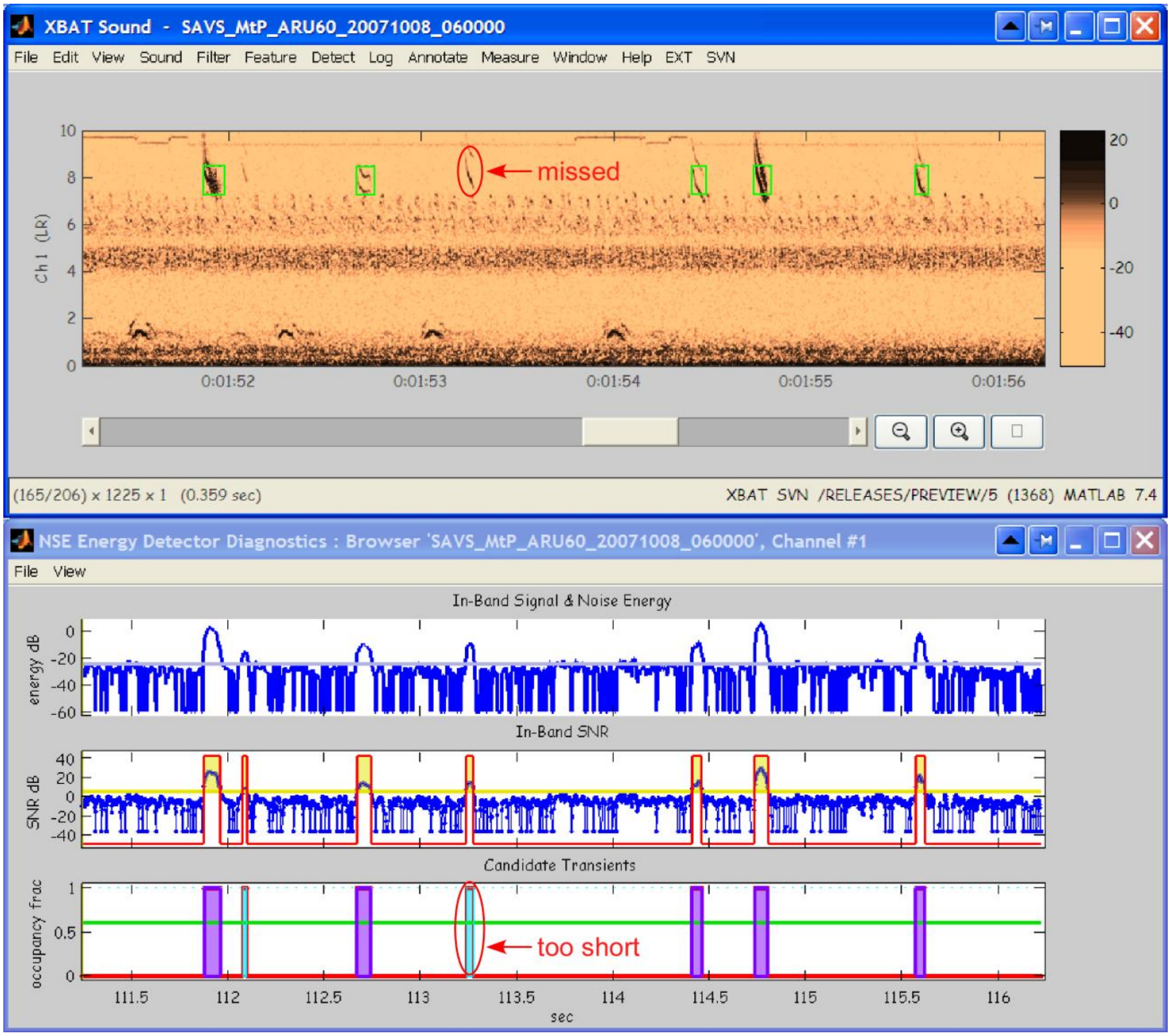

Figure 3. Band limited energy detector diagnostic display in XBAT. **Upper window:** Sound spectrogram of five seconds of a night-time recording made by an ARU, showing six nocturnal flight calls of savannah sparrow (*Passerculus sandwichensis*). Green rectangles mark five calls that were detected by the energy detector. The red ellipse marks a call that was missed by the detector. **Lower window:** Energy detector diagnostic display. The turquoise highlighting of the third candidate transient (bottom panel) indicates that this candidate, which corresponds to the missed detection in the upper window, was rejected because it did not satisfy the minimum duration criterion specified in the detector configuration.

Database logs in XBAT

Each sound recording that is analyzed in XBAT may have one or more *logs* associated with it. A log stores information about *events*. Each event has associated with it a time (where it is in the recording), a duration, and a minimum and maximum frequency. Events may also have measurements and annotations (e.g., species tags) associated with them. Events can be created

manually by a user (by drawing time-frequency boxes on a spectrogram) or automatically by a detector.

At the outset of this project, logs were saved in the form of MATLAB data (.mat) files, which was the most natural and convenient way to store the data structures representing events. However, as experience accumulated using a variety of detectors on very large datasets, limitations of this storage format became apparent. In order to work with a log (e.g., for a user to review events logged by a detector), XBAT had to read the entire log file into memory. As log size increased beyond approximately 10,000 events the performance of the system for even simple tasks (e.g., paging from one event to the next) became unacceptably slow. In projects involving very long recordings, detection runs sometimes generated logs with many tens of thousands of events that were effectively unusable. Although workarounds were possible (e.g., running detections on tiled subsets of the data, creating a series of smaller logs) the "large log problem" became the overall limiting factor on the rate at which data could be processed.

To address the problems created by large event logs, work was undertaken under SI-1461 to implement a new storage format for XBAT event logs as SQLite databases. Using a database representation would enable fast access to arbitrarily large logs, independent of log size, thus overcoming the bottleneck posed by large MATLAB-format logs. A database representation also has the added advantage that XBAT logs would become readily accessible from outside the XBAT or MATLAB environments, as they could be searched via SQL queries, either directly by a human user or by programs written in a wide variety of other programming languages. The original MATLAB-format logs were inaccessible from outside of MATLAB without major programming efforts.

Implementation of database XBAT logs occurred in two phases. In the first (infrastructure) phase, MATLAB was extended via the MEX interface to support read-write access to SQLite database files. In the second phase, a database schema for storage of event log information was developed and implemented as part of XBAT. As a result of this work, database logs are now fully functional in the development version of XBAT. Tests have verified that database logs containing hundreds of thousands of events can now be used with rapid access to all data, at speeds indistinguishable from logs containing hundreds of events. Logs of this size stored in the older MATLAB format would have been impossible to use. This development marks a major improvement in the usability of XBAT for processing the amounts of data typically acquired by large-scale passive acoustic monitoring of bird habitats.

Nearest-neighbor classifier in XBAT

Nearest-neighbor (NN) classification is an established instance-based machine learning method (Cover and Hart 1967). To classify unknown instances, it relies on an existing library of labeled training examples and a distance-based notion of object similarity. A distance function is used to determine which labeled examples are closest, and therefore assumed most similar, to the unlabeled object. The object's label is then predicted by the label of either the single nearest neighbor, or a through a voting rule on the collection of *k* nearest neighbors (*k*-NN). Object distances are calculated from measured object features using a choice of distance functions. For sound data, it is common to generate distances via spectrogram cross-correlation (e.g., Clark et al. 1987, Cortopassi and Bradbury

2000), essentially using the entire spectrogram as a feature vector. Distances can also be computed through a variety of extracted measurements (features).

NN classification is straightforward and theoretically effective as training sets become larger (Cover and Hart 1967). However, searching through the library of examples to find neighbors becomes time consuming, as the training set grows in size and many features are considered to determine distance. For bioacoustic data, large example sets are often required to span the natural variation in acoustic signals. Furthermore, typical distance metrics like spectrogram correlation present an enormous number of feature dimensions.

Under SI-1461, work was done on two fronts to support implementation of practical NN classification tools.

Implementation of condensed nearest-neighbor domain description (CNNDD)

Ideally a set of sound examples for a NN classifier should be large enough to span the range of biological variability, but no larger, because increasing the number of examples to be searched increases processing time. Typically, an arbitrarily compiled set of examples—e.g., all known examples of a particular sound type from an archive such as the Macaulay Library—is highly redundant and much larger than necessary. In order to speed processing, it would be desirable to use only a subset of all available examples, chosen so that the subset is as small as possible while still spanning the natural range of variation. The condensed nearest-neighbor domain description (CNNDD) algorithm (Angiulli 2007) is a method for finding a subset of a large, redundant example set such that, when the subset is used with a particular NN classifier, it will provide classification performance equivalent to that of the complete example set.

Under SI-1461, the CNNDD algorithm was implemented in MATLAB in a form that can be readily integrated to support NN classification in XBAT. Table 4 illustrates the performance of the CNNDD algorithm as implemented with data on nocturnal flight calls from four species of warbler. These data were collected as part of DoD Legacy Project 5-245.

Table 4. Condensation of exemplar sets for nearest-neighbor classification of migratory nocturnal flight calls of four species of warblers. 'Total exemplars' is the number of exemplars available in the complete unreduced set of known sounds. The last two columns show numbers of exemplars in the condensed sets for equivalent nearest-neighbor classification performance with 1 or 5 nearest neighbors. AMRE = American redstart (*Setophaga ruticilla*), COYE = common yellowthroat (*Geothlypis trichas*), OVEN = ovenbird (*Seiurus aurocapilla*), MAWA = magnolia warbler (*Dendroica magnolia*).

		Condensed exemplars	
Species	**Total exemplars**	**1 nearest neighbor**	**5 nearest neighbors**
AMRE	379	25	46
COYE	89	8	16
OVEN	459	14	24
MAWA	1872	17	30

Implementation of metric trees for fast nearest neighbor classification

Metric trees are data structures that are used to organize and search large sets of data in a multidimensional "metric space" by recursively partitioning that space into successively smaller volumes by a series of hyperplanes. Metric trees can be used to rapidly find the nearest neighbor to an object of unknown identity in a set of labeled example objects.

Figure 4 illustrates this process with a set of 64 example objects in a simple hypothetical two-dimensional feature space (Figure 4a; real acoustic data would be represented with a much larger number of dimensions than can easily be shown in a two-dimensional illustration). In Figure 4b, the algorithm has created a metric tree spanning the feature space, with nodes indicated by small white circles. The root of the tree (the starting point for traversing the tree to classify an unknown object) is in the center, marked by a bold black border. Each terminal node, or *leaf* of the tree corresponds to a neighborhood of eight example objects; the actual number of examples associated with each leaf is a configurable parameter of tree construction. Also shown in Figure 4b is an object of unknown type to be classified (indicated by red 'X' in the small yellow circle). To classify the unknown, the object is compared to individual nodes in the tree, beginning with the root. Each comparison determines whether the unknown is to the left or the right of a line (not shown) through the node perpendicular to the tree. (In an *n*-dimensional space, the comparison determines which side of an *n*-1-dimensional hyperplane the unknown is on.) The comparison then moves to the next node down the tree in the chosen direction (Figure 4c, red arrow). In this way the tree is traversed until the leaf nearest to the unknown is reached (Figure 4d). The distance between the unknown and each of the examples associated with the chosen leaf is then computed to find the nearest example. The class (species) label of that example is then assigned to the unknown. In this approach, the number of comparisons that need to be made increases as the logarithm (base 2) of the number of examples. So, for example a million examples could be searched with approximately 20 comparisons.

Under SI-1461, a metric tree construction algorithm (Liu et al. 2004) was implemented in MATLAB in a form that can be readily integrated to support NN classification in XBAT.

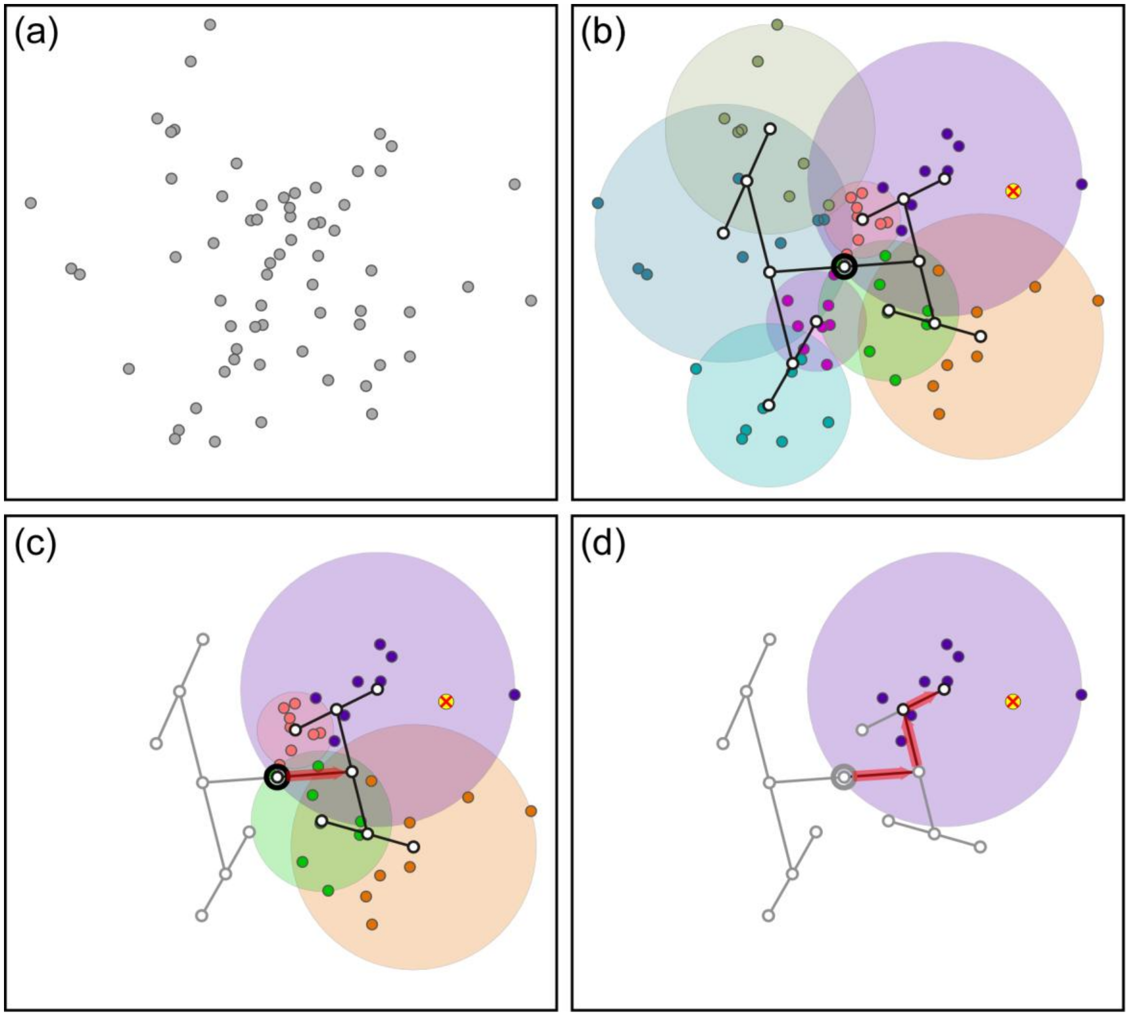

Figure 4. Conceptual illustration of fast nearest-neighbor searching using metric trees in a 2-dimensional space. In the case of real acoustic data, the space to be searched would have many dimensions, with each dimension corresponding to an acoustic feature. **(a)** A set of 64 known examples of the types of objects to be classified. Each object is labeled with a class ("species"), not shown. **(b)** A metric tree constructed to represent the example set. Nodes of the tree are indicated by white dots; the root of the tree is in the center, indicated by the node with a bold black border. Each terminal node or leaf is associated with a set of eight examples. Large colored circles indicate the range between each terminal node and its most distant example. The circled red 'X' represents an unknown object to be classified. **(c)** First step in classification of the unknown object by traversing the tree. Examples associated with the left side of the tree have been eliminated from consideration. **(d)** Final step in traversal of the tree. The domain of examples to which the unknown needs to be compared has been reduced to the eight examples associated with one leaf. Final classification is done by evaluating the distance between the unknown and each of the eight remaining examples, then assigning the class label of the nearest one.

NFC detector infrastructure and plug-ins for nocturnal flight call monitoring in Raven

Raven plug-in architecture and detector infrastructure

Under SI-1461, the architecture used in Raven 1.2.1 was enhanced to achieve a high level of modularity and extensibility. This redesign was motivated in large part by the goal of enabling developers to easily add new detection algorithms (such as those envisioned as part of SI-1461) to Raven without needing to be familiar with other parts of the application code. The resulting new version, Raven Pro 1.3, employs the Eclipse 3.1.2 plugin framework provided by the Eclipse Foundation (www.eclipse.org).[2] A plugin is a self-contained unit of code and/or data which may be independently added to a software application. Using this architecture, new features (such as new types of signal detectors) can be added to an existing installation of Raven Pro by simply placing a set of program files into the appropriate subdirectory within the Raven Pro directory; no recompiling or complex installation procedure is required. The Eclipse plugin framework serves a dual purpose: it assembles an application's constituent plugins into a working product and allows the plugins to be automatically updated from remote network locations.

Raven Pro now defines six classes of plugins, such as audio input devices, automatic detectors, and Fourier Transform algorithms. In total, Raven Pro is composed of 21 plugins, including four audio input devices and three automatic detectors. Users may install new plugins as they become available, installing only what is needed. This helps keep the user interface simple and easy to use. Maintenance is easy using the Eclipse automatic update facility.

In addition to modularity, Raven Pro's plugin framework allows software developers outside the Cornell Lab of Ornithology to extend the capabilities of Raven Pro by contributing plugins to the project. Developers may write extensions in Java. The automatic detector plugin class also has a facility for writing detectors in Python.

Band-limited energy detector plug-in for Raven Pro

Under SI-1461, a band-limited energy detector (BLED) plug-in was implemented using Raven Pro's new plugin architecture. The algorithm used in this detector is similar to that described above for the XAT energy detector. To use the detector, the user specifies minimum and maximum frequency and duration, and minimum separation in time for events of interest, as well as parameters for background noise estimation (Figure 5). The detector identifies events for which the estimated in-band energy exceeds the background noise estimate by the specified SNR for a duration within the specified limits. Figure 6 shows an example of bird calls identified by the BLED.

[2] Eclipse is an open source project managed by a consortium of corporations which includes Hewlett-Packard, IBM, and Intel.

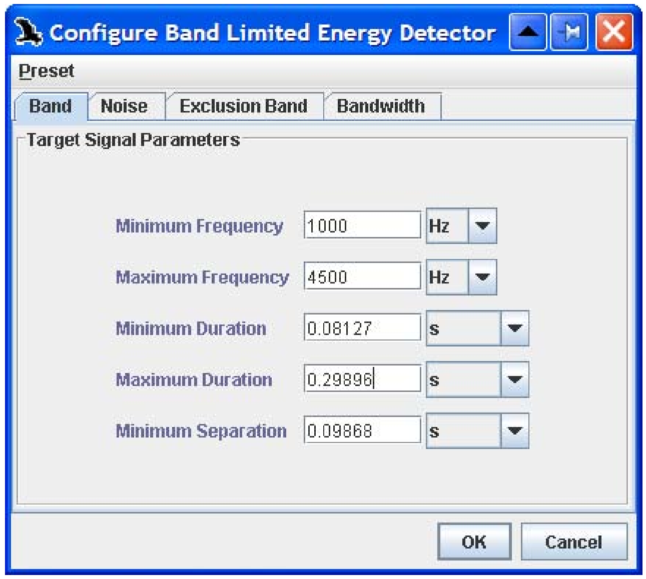

Figure 5. The configuration dialog box for Raven's band-limited energy detector.

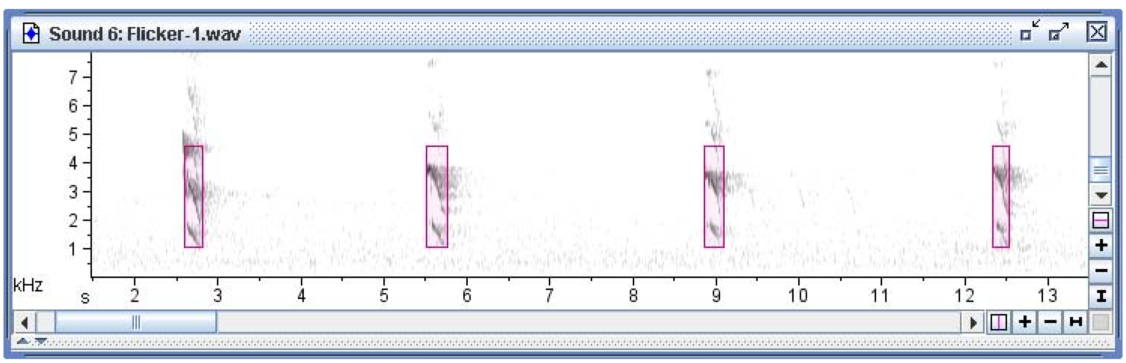

Figure 6. Calls of a Northern Flicker (*Colaptes auratus*) detected and highlighted by Raven's band-limited energy detector.

The BLED is now being used by the DoD Legacy migratory bird monitoring project to detect nocturnal flight calls of migrant birds passing over the bases listed in Table 1. The BLED typically processes these recordings at over 200 times real-time speed, so that an eight-hour recording is completely processed in slightly more than two minutes. Raven Pro allows the user to run multiple detectors at once on the same or different data sets. In addition, Raven Pro can run the same detectors in real-time on live data streamed from a microphone. This capability is crucial to the use of Raven as a real-time monitoring tool as envisioned in the SI-1461 proposal.

These advances in Raven's architecture and detection capabilities are already being exploited by the companion DoD-funded Legacy project applying acoustic technologies to studies of migrating birds. This project has made extensive use of software to automate processing of tens of thousands of hours of recordings, focusing on band-limited energy detectors as a means to extract flight calls and other vocalizations of interest as rapidly as possible. Raven Pro has been used (1) to detect signals of interest (primarily flight calls), (2) to extract (export) these signals as clips for later viewing and analysis

(classification), and (3) to view these signals of interest. Raven Pro's high processing rate (> 200x real-time), combined with its ability to handle very large selection tables (on the order of 100,000-300,000 events or more) have enabled the Legacy researchers to proceed with verifying these data for valid flight calls much more efficiently than ever before. These advances have enabled the Legacy team to begin to develop data analysis procedures operable on a scale commensurate with our ability to collect massive amounts of data at low cost using ARU technology.

Future directions

The original proposal for this project identified three major areas of development to be undertaken over the course of four years: (1) detection and classification software to support long term acoustic monitoring using ARUs, (2) enhancements to balloon-based acoustic monitoring, and (3) nocturnal flight call monitoring. The developments described above represent slightly more than one year of effort in areas 1 and 3.

This section describes directions that should be taken by future efforts in these areas.

Detection and classification software

In recent years, there has been increasing interest in developing automated, quantitative methods for classifying acoustic signals of animals. Multiple classification techniques have shown promising results including, for example, artificial neural networks (Murray et al. 1998, Deecke et al. 1999, Deecke et al. 2000, Parsons and Jones 2000, Dawson et al. 2006, Nickerson et al. 2006, Selin 2007), hidden Markov models (Kogan and Margoliash 1998, Skowronski and Harris 2006, Chen and Maher 2006, Somervuo et al. 2006), template matching with dynamic time warping (Anderson et al. 1996; Kogan and Margoliash 1998, Somervuo et al. 2006), Gaussian mixed models (Skowronski and Harris 2006, Somervuo et al. 2006, Kwan et al. 2006, Roch et al. 2007), discriminant function analysis (Cortopassi and Bradbury 2000; Parsons and Jones 2000, Kazial et al. 2001, Lee et al. 2006, Skowronski and Harris 2006), and classification and regression trees (Melendez et al. 2006). These approaches have been applied to signals from a variety of taxa including birds (Anderson et al. 1996, Kogan and Margoliash 1998, Cortopassi and Bradbury 2000, Chen and Maher 2006, Dawson et al. 2006, Nickerson et al. 2006, Somervuo et al. 2006, Selin 2007), bats (Parsons and Jones 2000, Kazial et al. 2001, Melendez et al. 2006, Skowronski and Harris 2006), odontocetes (Hayward 1997; Murray et al. 1998; Deecke et al. 1999; Houser et al. 1999; Roch et al. 2007), terrestrial mammals (Placer and Slobodchikoff 2000), anurans (Lee et al. 2006), and insects (Chesmore 2001, Chesmore and Ohya 2004, Lee et al. 2006).

As important as the choice of classifier (or perhaps more so) is the choice of features (measurements) to be extracted and provided as input to the classification algorithm. Many bird sounds, particularly the advertising songs of passerines, are characterized by variable and complex hierarchical structures of simple subunits. Many of the studies cited above rely on various types of low-level spectral measurements that fail to capture this higher-level structure. Future work should include efforts to identify higher-level syntactical units (e.g., phrases of repeated or alternating subunits), patterns of which are often used by human experts to identify bird sounds.

Open-mic recordings in natural environments (such as those made with ARUs) typically include multiple signal sources, often overlapping in time and frequency, which increases the difficulty of classification and detection. Future efforts should include exploration of blind source separation techniques such as independent component analysis (Hyvärinen et al. 2001), which can aid in isolating sounds to be classified.

Balloon-based acoustic monitoring

The original balloon recording system developed under SI-1185 used two microphones suspended from the ends of a 1 m long horizontal boom. This system should be replaced with a pair of microphones suspended beneath the balloon, with a few meters of vertical separation, as described in the SI-1461 proposal. The delay in the arrival of each sound at the higher microphone, relative to the lower one, would be used in conjunction with the balloon's altitude to measure the distance from the singing bird to the point on the ground directly beneath the balloon. This measure of distance enables subsequent analyses to estimate how detection probability falls off with distance, and thus estimate the area surveyed for each species.

The balloon system's altitude control system should be upgraded to address two issues. First, the current software sometimes overcompensates for rapid altitude changes caused by a combination of higher wind speeds and steep terrain, which can lead to loss of the system (via either premature landing or excessive altitude gain). These problems could be addressed by enhancing the software to ignore rapid altitude changes likely to be caused by terrain-following winds. Second, the current system begins its programmed descent (by venting helium) only once the balloon crosses the defined perimeter of the search area. The maximum rate at which the current valve design can vent sometimes leads to undesirably long descents, and landings far outside the target perimeter, which can hamper recovery efforts. We would address this by (a) increasing the maximum orifice of the valve to allow for faster venting, and (b) revising the software to initiate the descent phase before crossing the boundary, to target a landing closer to the boundary.

Additional changes should be made to communication between the balloon in flight and personnel on the ground, in order to improve the efficiency of instrument recovery upon conclusion of a flight.

Nocturnal flight call monitoring

In addition to the implementation of detector infrastructure and a prototype detector for Raven (completed, as described above), the original proposal identified the following software and hardware development tasks associated with nocturnal flight call monitoring:

1. an acoustic database to host nocturnal flight call audio clips uploaded from a network of monitoring stations;
2. client software on monitoring computers to upload detected sounds to the database;
3. NFC classification algorithms;
4. prototype NFC detection network

At the time the proposal for SI-1461 was written, no network-accessible acoustic database existed, so the development of this resource was a key item in the proposal. However, as a result of recent work

in our laboratory funded by other sources, we now have two such databases, implemented as part of the Right Whale Listening Network and the Bioacoustic Resource Network, either of which could potentially be adapted to form the hub of a NFC monitoring network. These two systems are summarized below.

The Right Whale Listening Network[3] was developed to provide near-real-time acoustic detection of endangered North Atlantic right whales (*Eubalaena glacialis*) in and near the commercial shipping lanes approaching Boston, Massachusetts. The system, which has been online in continuous operation since January 2008, was developed to warn commercial shipping vessels of the presence of right whales in the area, in order to mitigate the hazard of ship-whale strikes, which are a major cause of mortality for this highly endangered whale species. The offshore portion of the network consists of a set of “auto-detection buoys” each equipped with an underwater microphone (hydrophone), onboard signal detection software, and satellite communication system. At programmed intervals (currently every 20 minutes) each buoy communicates via a satellite link with a database server in our lab, and uploads clips of possible right whale sounds that it has detected. The server supports a password-protected website through which authorized expert users can review and validate uploaded sound clips, and a separate outreach site (http://www.listenforwhales.org/, Figure 7) where the general public can view near-real-time reports on where whales have been detected by the network in the past 24 hours.

[3] The Right Whale Listening Network was developed by the Cornell Bioacoustic Research Program and the Woods Hole Oceanographic Institution with funding from non-SERDP Federal and Massachusetts state agencies and industry partners, in cooperation with several NGOs.

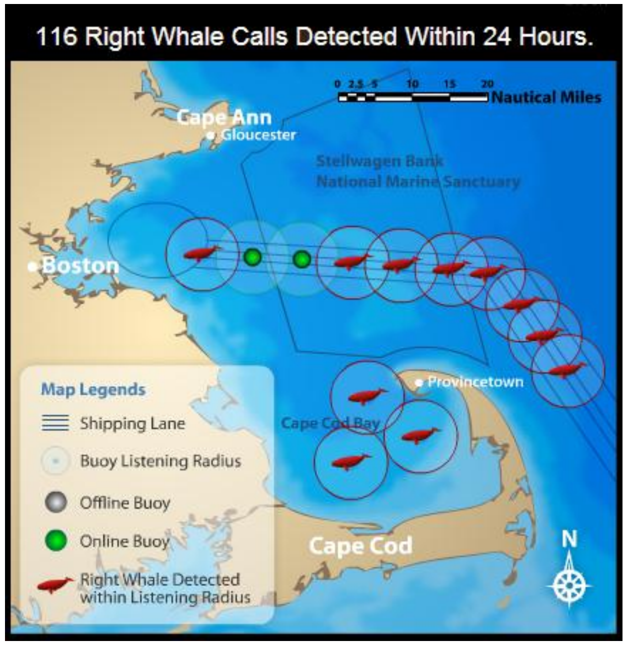

Figure 7. Real-time status map for the Right Whale Listening Network from the public website (http://www.listenforwhales.org/). Red whale icons show positions of acoustic monitoring buoys that have detected right whale calls within the past 24 hours; small green circles indicate operational buoys without detections.

The BioAcoustic Resource Network (BARN, http://barn.xbat.org, Figure 8) consists of an Internet-accessible acoustic database and associated software tools to support collaborative bioacoustic research and monitoring projects. BARN's database infrastructure and network communication protocols are now in alpha testing. Because BARN uses established HTTP requests to control data transfers (e.g., HTTP POST to upload a sound clip), implementing a client for uploading flight call clips would be a simple task in any modern programming language (e.g., Java, Python), most of which have built-in support for HTTP communication. BARN is being implemented in conjunction with the XBAT project, and among the services that BARN will provide is server-side processing of sounds with any of the tools that are part of the XBAT core and extensions (e.g., detectors, classifiers). Thus, once sound clips are uploaded to BARN by nodes in the NFC monitoring network, they could be classified by software running on the server, and the results could be made available over the Internet to authorized users anywhere via a web-browser interface. Users could view and listen to sound clips and see the classifications proposed by the system (Figure 8). They could validate the proposed classifications, or edit them based either on their own expert knowledge or on comparisons to a library of calls of known identity, which would be made available by the BARN system. A BARN-based system could be used by authorized experts to validate machine classifications, or could form the basis of a citizen-science project that would recruit large numbers of volunteer users to bring human pattern-recognition skill to the task of validating classifications, similar to the Cornell Lab of Ornithology's CamClickr project (http://watch.birds.cornell.edu/nestcams/clicker/clicker/index), which uses citizen scientists to classify and tag millions of images of bird behavior at nests.

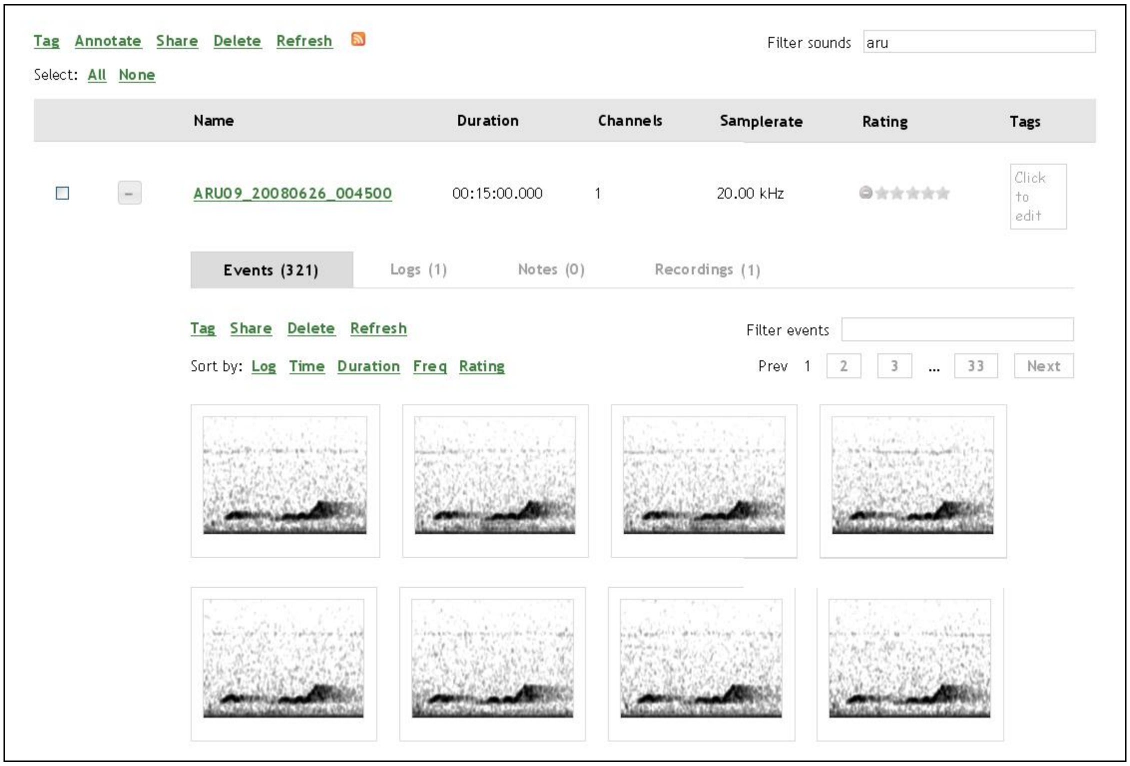

Figure 8. Screenshot of the BioAcoustics Resource Network (BARN) website (http://barn.xbat.org) showing the interface for reviewing events in a log associated with a sound. In this example, each small spectrogram shows a single phrase from a whip-poor-will song, detected by the data template detector. From this page, a user can play any sound (by clicking on its spectrogram image), or can apply tags, ratings, and annotations.

Either the right whale listening network or BARN could potentially provide much of the database and network communication infrastructure required for a nocturnal flight call monitoring network. If SERDP or another source were to fund further development of such a network, a first step would be a more in-depth evaluation of which of these would provide a more appropriate foundation, depending on more detailed consideration of the needs of an NFC network, and the development state of these two projects at the time. In either case, much of the work necessary for developing the NFC network has now been done with non-SERDP funding.

Deployment of the necessary hardware (directional microphones and preamplifiers) for a prototype/demonstration network, as described in the proposal, remains to be done.

Further work on classification of nocturnal flight calls is needed, and is underway presently in our lab (with funding from other source), building on the progress described in this report.

Acknowledgements

The diagnostic displays for the XBAT band-limited energy detector were developed and implemented by Kathryn Cortopassi. XBAT and the data template detector were developed by Harold Figueroa, with assistance from Matt Robbins. Improvements to Raven Pro described here were designed,

implemented, tested, and documented by Dean Hawthorne, Dounan Hu, Andrew Jackson, Tim Krein, Harold Mills, Scott Nelson, and Laura Strickman.

ARU recordings used for detector development and testing were acquired through the efforts of Melanie Driscoll, Andrew Farnsworth, Stefan Hames, Mike Powers, and Scott Stoleson (USDA Forest Service). Chris Tessaglia-Hymes provided ARU technical and logistical support. Mike Pitzrick performed the evaluations of detector performance on recordings of Cerulean Warbler and Whip-poor-will. Russell Charif directed the analysis of those recordings, and prepared this report.

Portions of the work described here were partially supported by DoD-Legacy and by USDA Forest Service in addition to SERDP.

Literature cited

Alldredge, M. W., K. H. Pollock and T. R. Simons. 2006. Estimating detection probabilities from multiple-observer point counts. *Auk* 123(4): 1172-1182.

Anderson, S. E., A. S. Dave and D. Margoliash. 1996. Template-based automatic recognition of birdsong syllables from continuous recordings. *J. Acoust. Soc. Am.* 100(2): 1209-1219.

Angiulli, F. 2007. Condensed nearest neighbor data domain description. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(10): 1746-1758.

Bardeli, R., D. Wolff and M. Clausen. 2007. Bird song recognition in complex audio scenes. *Computational Bioacoustics for Assessing Biodiversity*. Isle of Vilm, Germany, BfN (Federal Agency for Nature Conservation, Germany).

Chen, Z. X. and R. C. Maher. 2006. Semi-automatic classification of bird vocalizations using spectral peak tracks. *Journal of the Acoustical Society of America* 120(5): 2974-2984.

Chesmore, E. D. 2001. Application of time domain signal coding and artificial neural networks to passive acoustical identification of animals. *Applied Acoustics* 62(12): 1359-1374.

Chesmore, E. D. and E. Ohya. 2004. Automated identification of field-recorded songs of four British grasshoppers using bioacoustic signal recognition. *Bulletin of Entomological Research* 94(4): 319-330.

Clark, C. W., P. Marler and K. Beeman. 1987. Quantitative analysis of animal vocal phonology: an application to swamp sparrow song. *Ethology* 76: 101-115.

Cortopassi, K. A. and J. W. Bradbury. 2000. The comparison of harmonically rich sounds using spectrographic cross-correlation and principal coordinates analysis. *Bioacoustics* 11: 89-127.

Cover, T. M. and P. E. Hart. 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory* 13(1): 21-+.

Dawson, M. R. W., I. Charrier and C. B. Sturdya. 2006. Using an artificial neural network to classify black-capped chickadee (*Poecile atricapillus*) call note types. *Journal of the Acoustical Society of America* 119(5): 3161-3172.

Deecke, V. B., J. K. B. Ford and P. Spong. 1999. Quantifying complex patterns of bioacoustic variation: Use of a neural network to compare killer whale (*Orcinus orca*) dialects. *Journal of the Acoustical Society of America* 105(4): 2499-2507.

Deecke, V. B., J. K. B. Ford and P. Spong. 2000. Dialect change in resident killer whales: implications for vocal learning and cultural transmission. *Animal Behaviour* 60: 629-638.

Hayward, T. J. 1997. Classification by multiple-resolution statistical analysis with application to automated recognition of marine mammal sounds. *Journal of the Acoustical Society of America* 101(3): 1516-1526.

Houser, D. S., D. A. Helweg and P. W. Moore. 1999. Classification of dolphin echolocation clicks by energy and frequency distributions. *Journal of the Acoustical Society of America* 106(3): 1579-1585.

Hyvärinen, A., J. Karhunen and E. Oja. 2001. *Independent Component Analysis*, Wiley-Interscience.

Kazial, K. A., S. C. Burnett and W. M. Masters. 2001. Individual and group variation in echolocation calls of big brown bats, *Eptesicus fuscus* (Chiroptera : Vespertilionidae). *Journal of Mammalogy* 82(2): 339-351.

Kogan, J. A. and D. Margoliash. 1998. Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study. *Journal of the Acoustical Society of America* 103(4): 2185-2196.

Kwan, C., K. C. Ho, G. Mei, Y. Li, Z. Ren, R. Xu, Y. Zhang, D. Lao, M. Stevenson, V. Stanford and C. Rochet. 2006. An automated acoustic system to monitor and classify birds. *Eurasip Journal on Applied Signal Processing*.

Lee, C. H., C. H. Chou, C. C. Han and R. Z. Huang. 2006. Automatic recognition of animal vocalizations using averaged MFCC and linear discriminant analysis. *Pattern Recognition Letters* 27(2): 93-101.

Liu, T., A. W. Moore, A. Gray and K. Yang. 2004. *An Investigation of practical approximate nearest neighbor algorithms*. NIPS-2004. Eighteenth Annual Conference on Neural Information Processing Systems.

Melendez, K. V., D. L. Jones and A. S. Feng. 2006. Classification of communication signals of the little brown bat. *Journal of the Acoustical Society of America* 120(2): 1095-1102.

Mills, A. M. 1986. The influence of moonlight on the behavior of goatsuckers (Caprimulgidae). *Auk* 103(2): 370-378.

Murray, S. O., E. Mercado and H. L. Roitblat. 1998. The neural network classification of false killer whale (Pseudorca crassidens) vocalizations. *Journal of the Acoustical Society of America* 104(6): 3626-3633.

Nickerson, C. M., L. L. Bloomfield, M. R. W. Dawson and C. B. Sturdy. 2006. Artificial neural network discrimination of black-capped chickadee (*Poecile atricapillus*) call notes. *Journal of the Acoustical Society of America* 120(2): 1111-1117.

Parsons, S. and G. Jones. 2000. Acoustic identification of twelve species of echolocating bat by discriminant function analysis and artificial neural networks. *Journal of Experimental Biology* 203(17): 2641-2656.

Placer, J. and C. N. Slobodchikoff. 2000. A fuzzy-neural system for identification of species-specific alarm calls of Gunnison's prairie dogs. *Behavioural Processes* 52(1): 1-9.

Roch, M. A., M. S. Soldevilla, J. C. Burtenshaw, E. E. Henderson and J. A. Hildebrand. 2007. Gaussian mixture model classification of odontocetes in the Southern California Bight and the Gulf of California. *Journal of the Acoustical Society of America* 121(3): 1737-1748.

Selin, A., J. Turunen, . Tanttu. 2007. Wavelets in recognition of bird sounds. *EURASIP Journal on Advances in Signal Processing* 2007.

Skowronski, M. D. and J. G. Harris. 2006. Acoustic detection and classification of microchiroptera using machine learning: Lessons learned from automatic speech recognition. *Journal of the Acoustical Society of America* 119(3): 1817-1833.

Somervuo, P., A. Harma and S. Fagerlund. 2006. Parametric representations of bird sounds for automatic species recognition. *IEEE Transactions on Audio Speech and Language Processing* 14(6): 2252-2263.

Wilson, M. D. and B. D. Watts. 2006. Effect of moonlight on detection of Whip-poor-wills: implications for long-term monitoring strategies. *Journal of Field Ornithology* 77(2): 207-211.